**Advanced Information Systems Research Program**
**"A High Performance Computing Framework for CMB Data Management"**
**Year 1 Annual Report**

**P.I. Julian Borrill**
**Computational Cosmology Center, Lawrence Berkeley Laboratory &**
**Space Sciences Laboratory, UC Berkeley.**

## Introduction

The Cosmic Microwave Background (CMB) provides a snapshot of the Universe as it was only 400,000 years after the Big Bang. Tiny fluctuations in the CMB temperature and polarization encode the fundamental parameters of cosmology and ultra-high energy physics. Precise measurements of these fluctuations require extremely large and complex datasets whose analysis necessarily involves leading-edge high performance computing resources. For the last decade, the DOE's National Energy Research Scientific Computing (NERSC) Center has been the major provider of such resources to the CMB data analysis community, currently providing several million CPU-hours per year to over 100 analysts from a dozen experiments, including a guaranteed annual allocation to the Planck satellite mission.

The goals of this project are to simplify user access to, and to improve the efficiency of the use of, such HPC systems. This involves two elements:

i. The data and task management framework (DTMF) - a web interface to simplify the generation and management of data analysis tasks across collaborations, including staging the necessary data from archival storage, building job parameter files and submission scripts, and ingesting the resulting data and its associated metadata.
ii. The on-the-fly simulation (OTFS) library - a library of CMB data simulation tools the break the IO bottleneck to scaling CMB analyses to the petascale inherent in the traditional simulate/write/read/analyze data sequence by replacing the redundant IO with simulation on demand.

## Progress Over The Last Year

**DTMF:** Significant progress has been made on developing the DTMF infrastructure, including deploying a server to host the web services, implementing grid-authentication through the NERSC LDAP server so the users can access it using their NERSC username/password. Figure 1 shows the login and welcome pages; on logging in the user is presented with a list of the applications they are allowed to run (experiment-specific tools being accessible only to members of that collaboration, as authenticated by their Unix group memberships on the NERSC systems).

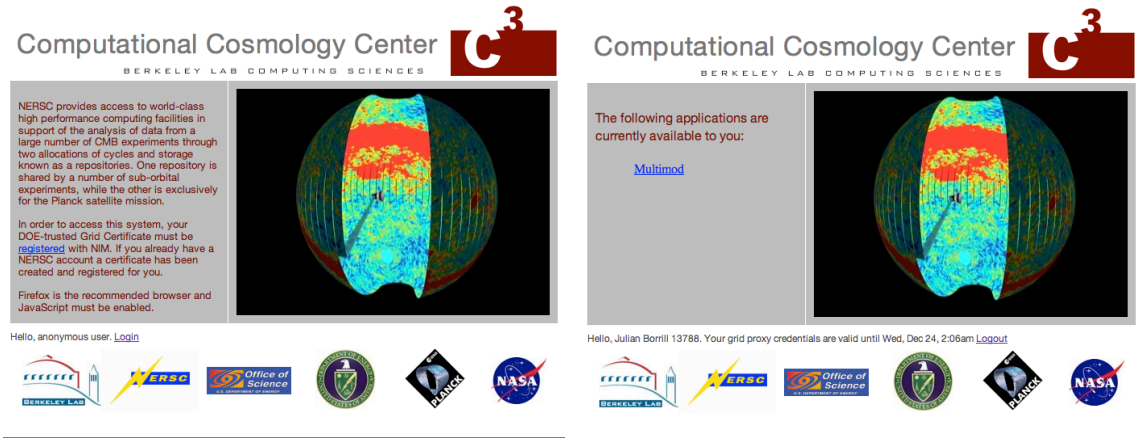


Figure 1: The DTMF login and welcome pages.

Based on the current workload, we have selected the Planck-specific time-ordered data simulation tool "multimod" as our first example application (figure 2).

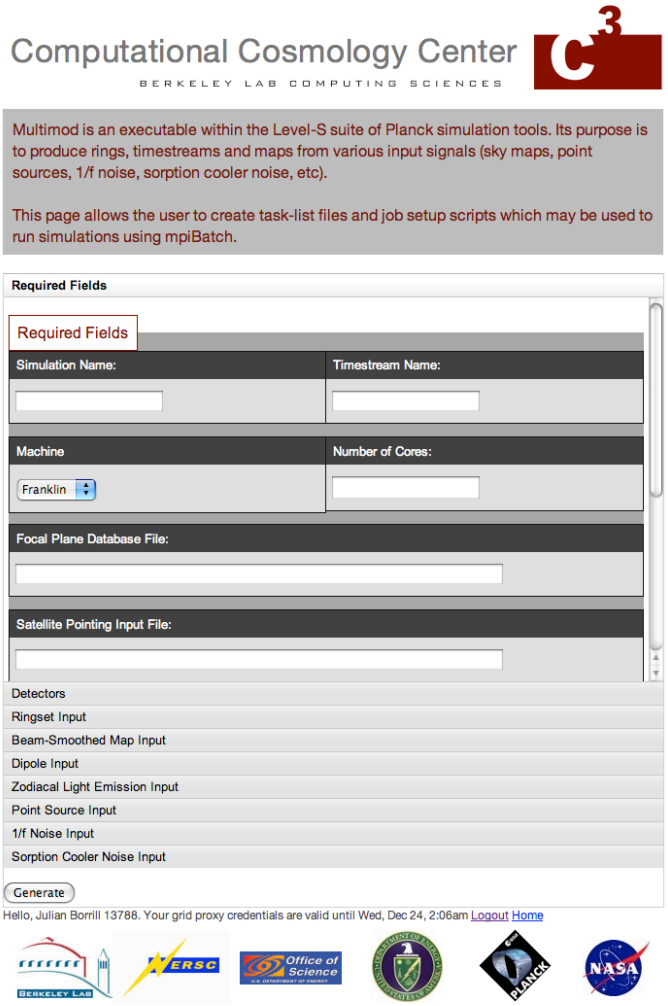


Figure 2: The DTMF multimod application page.

Within the thematically-grouped drop-down menu blocks the user defines the parameters of the run, including the overall simulation and particular component names, the target NERSC system and run concurrency, and all of the required parameters (multimod has approximately 50 possible command line arguments). The "generate" button then uses this information to build a list of multimod tasks (command lines), a system-specific PBS job submission script, and a setup shell script to build the appropriate directory structure and stage the necessary data for the simulation. The user then logs on to the target system, runs the setup script, and submits the job. This interface has been presented at the Planck Joint Core Team meeting in Bologna in November 2008, and will be presented to the US Planck Algorithm Development Group at its monthly face-to-face meeting in January 2009. As expected, there has been a lot of interest among users who have been performing Planck simulations by hand.

**OTFS:** The OTFS capability has also made dramatic progress over the last year.

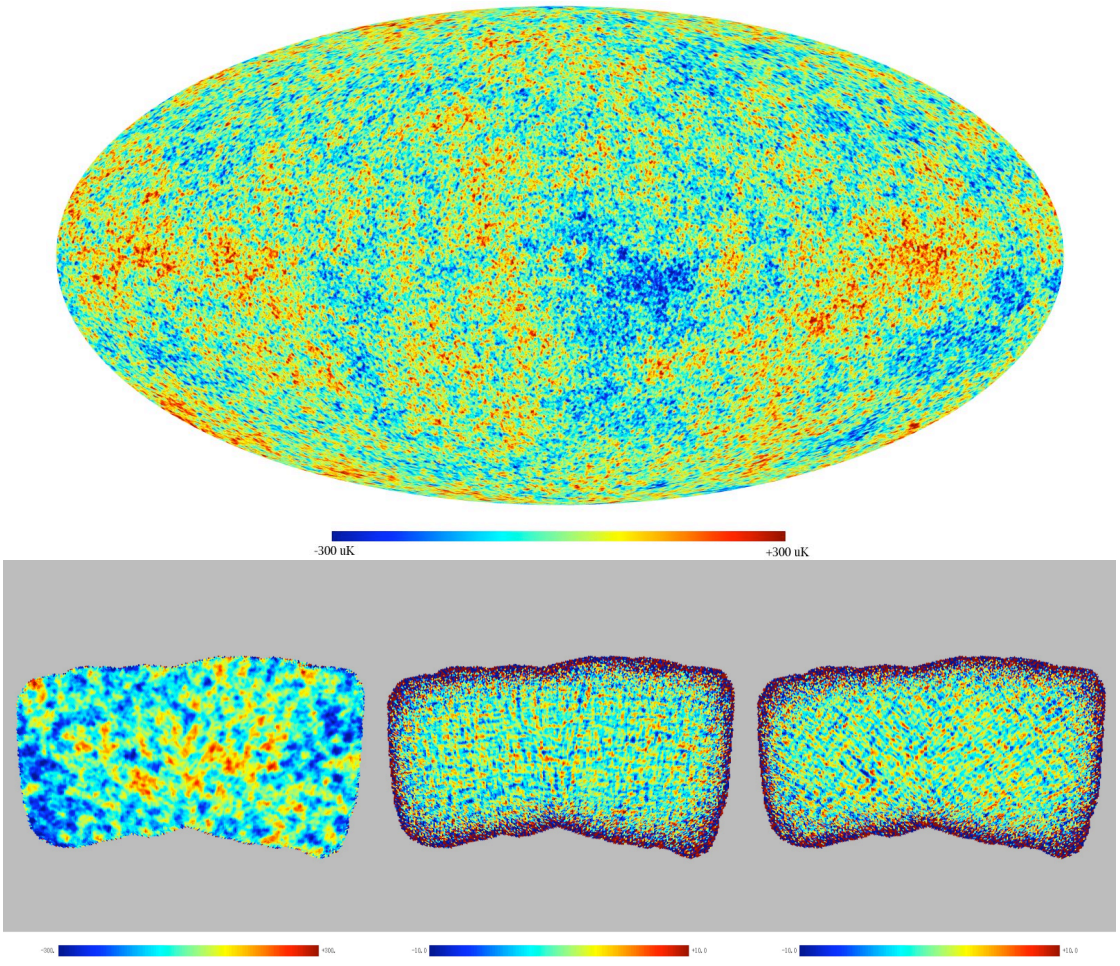


Figure 3: Planck 12-detector intensity and EBEX 720-detector intensity and Q- and U-mode polarization maps made with data simulated on the fly.

Using our M3 data abstraction layer, an analysis application code can now request time-ordered data that are to be simulated at runtime. Specifically, these data can include signals drawn from an input (possibly beam-smoothed) sky map, together with instrument noise with an arbitrary spectrum and stationary interval length. The challenges here have been the size of the input maps (larger than can be held in memory by a single process), and the reproducibility on any process of the random numbers used in the generation of any particular noise datum without having to generate the entire random sequence from scratch. These problems have been solved, and demonstrated with OTFS-based map-making runs on all 12 of the Planck 217 GHz detectors for 1 year, and 720 EBEX detectors at 3 frequencies for 2 weeks (figure 3).

These runs have also been performed at a range of concurrencies between 1,000 and 16,000 cores to demonstrate the improved scaling behaviour of the analysis, especially of the IO at very high concurrency. Figure 4 shows the total number of CPU-seconds spent on IO in three analyses - one with data read from disk and two using OTFS - where perfect scaling would be represented by a horizontal line. Clearly the OTFS library provides an enormous improvement in the scaling, and while it is still not perfect the IO is no longer an insurmountable bottleneck at high concurrency.

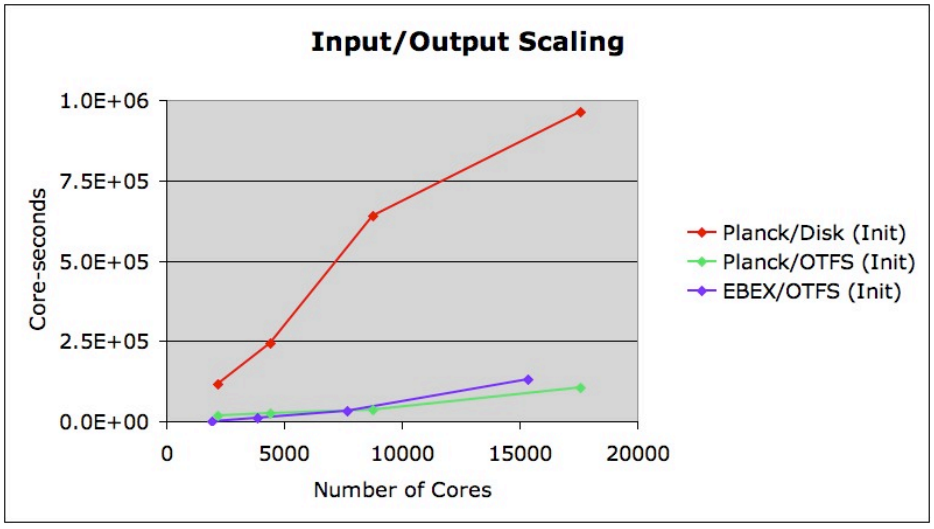


Figure 4: Comparative IO scaling for data read from disk and simulated on the fly.

This work has been included in presentations to a number of Planck and EBEX collaboration meetings in the fall/winter of 2008.

**Plans For The Coming Year**

Based on the progress made to date, we have identified the following goals for the coming year:

**DTMF:** The bulk of the effort in the coming year will be in extending DTMF. The priorities are

1. Add a tool-interface to build the run configuration files used by our M3 data abstraction layer, including for on-the-fly simulations.
2. Add a tool-interface to back-up mission-definition, time-ordered, and pixel domain data to archival storage, including ingesting their metadata into a database.
3. Begin the work of connecting the tool-interfaces to the database to be able to query the meta-data to locate possible data to be used in a particular analysis and to move whatever data is selected from archival to spinning storage, including its new location in the job description files.

**OTFS:** Extend the OTFS capability by allowing user-groups to provide their own simulation add-on tools, particularly for including experiment-specific systematic effects to simulated time-ordered data.